

Chat Bot through Google DialogFlow

1. Introduction :

A chatbot is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone.

1.1 Importance of ChatBot:

Chatbot is often described as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP).

1.2 Applications of ChatBot:

- Pizza Order.
- Product Suggestion
- Weather Report
- News and etc.,

2. Prerequisites:

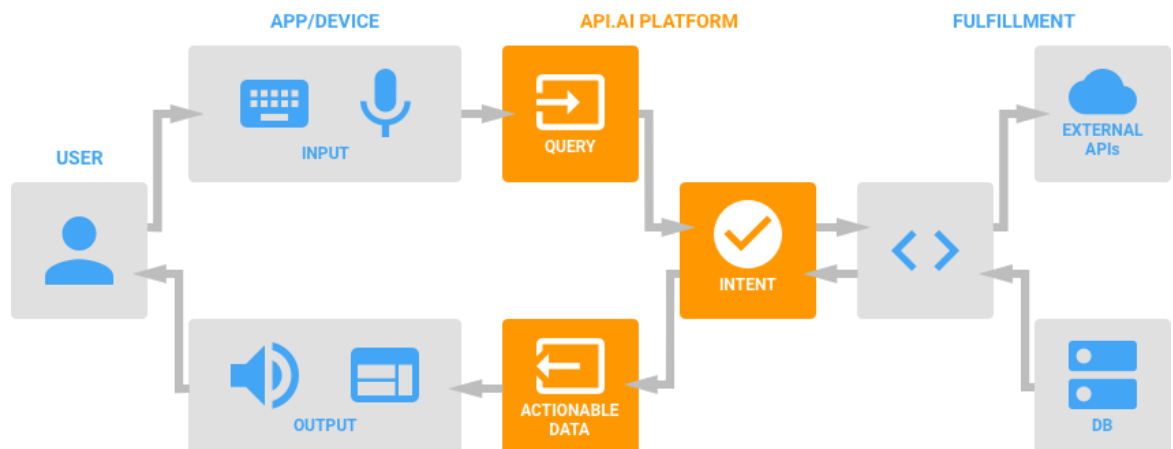
- Python – Version : 3.
- Google DialogFlow

3. Introduction to Google DialogFlow:

3.1 About Google DialogFlow:

Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems, and so on. Dialogflow incorporates Google's machine learning expertise and products such as Google Cloud Speech-to-Text.

3.2 Architecture of DialogFlow:



4. Problem Statement:

4.1 About:

The aim is to build a chat bot which answers the user queries about the number of corona virus cases within Indian states and all countries across the globe with its geographic location, about the corona virus, the symptoms and the preventive measure.

5. Solution – Approach:

5.1 Technical Stack:

- Python – version 3.6
- Google DialogFlow
- Microsoft cosmosDB – MongoAPI

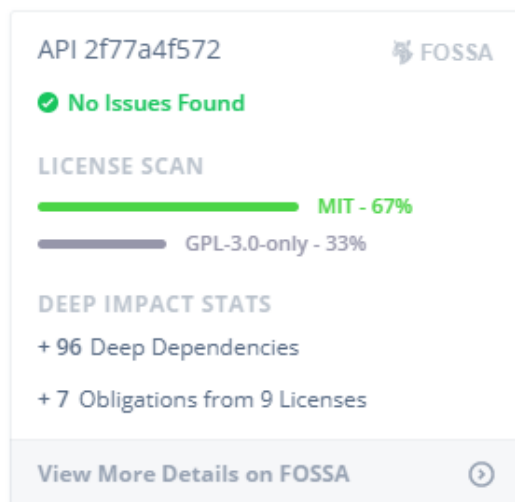
5.2 Libraries & Packages:

- Flask
- Pandas
- Requests
- Geopy

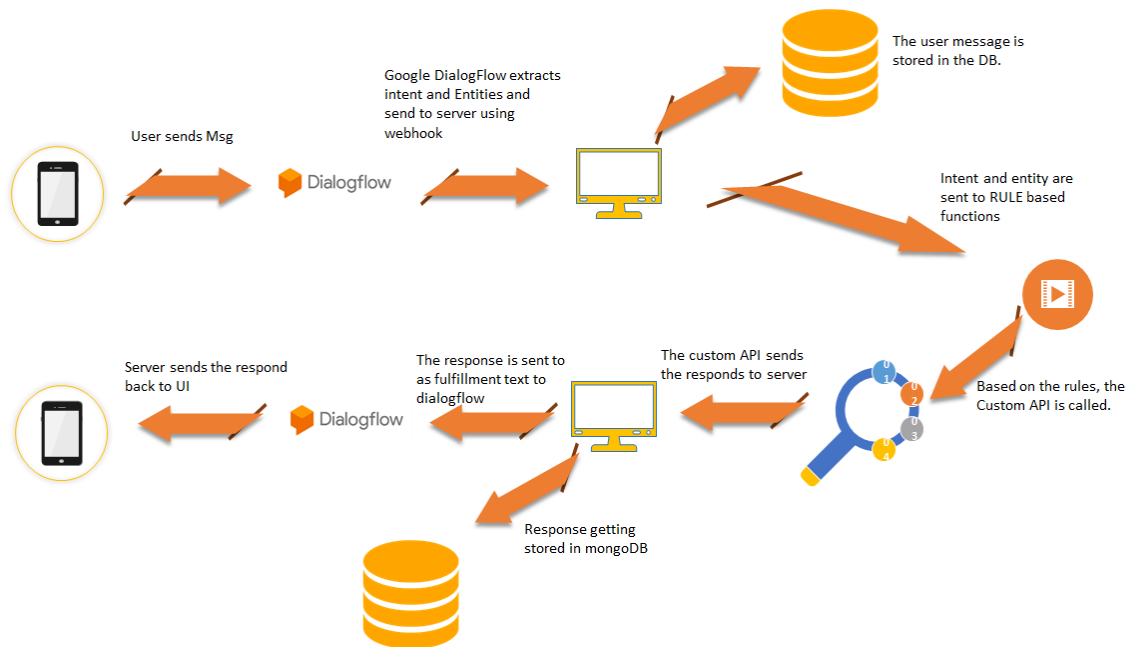
5.3 Third Party API's and License:

- API's used : <https://corona.lmao.ninja/docs/>
- Github : <https://github.com/NovelCOVID/API>
- Maps : mapQuest.com

License



5.4 Solution Architecture:



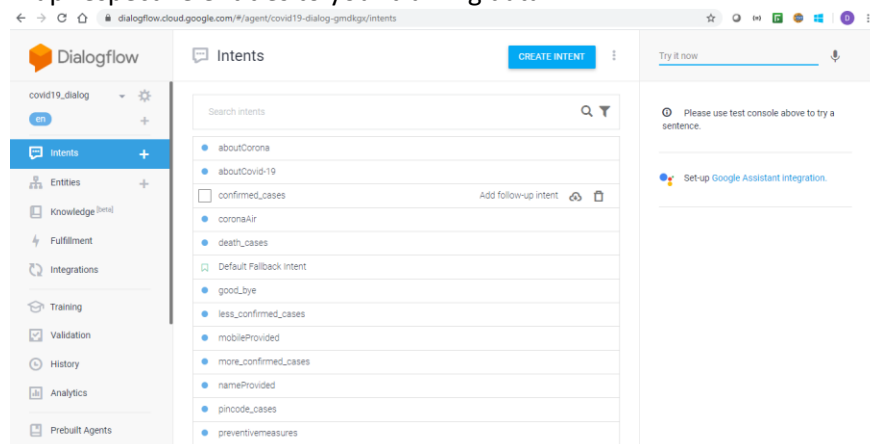
6. Implementation :

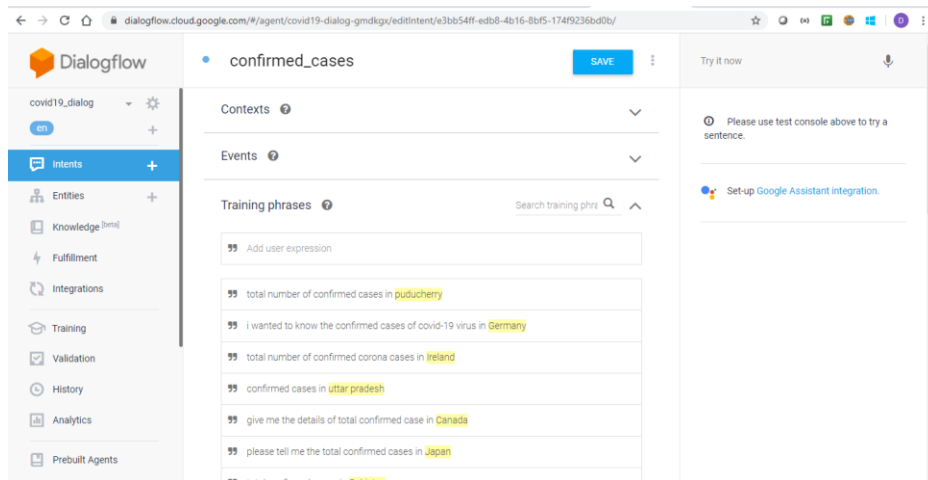
6.1 Library installation :

- Create a conda environment using “conda create -n <env_name>”
- Activate the conda environment.
- Pip install necessary libraries like Flask, pymongo, requests.

6.2 Training Data Creation in DialogFlow:

- In the google dialogflow create a new Agent.
- Under the intent section, create your own intents with the training phrases.
- Map respective entities to your training data.

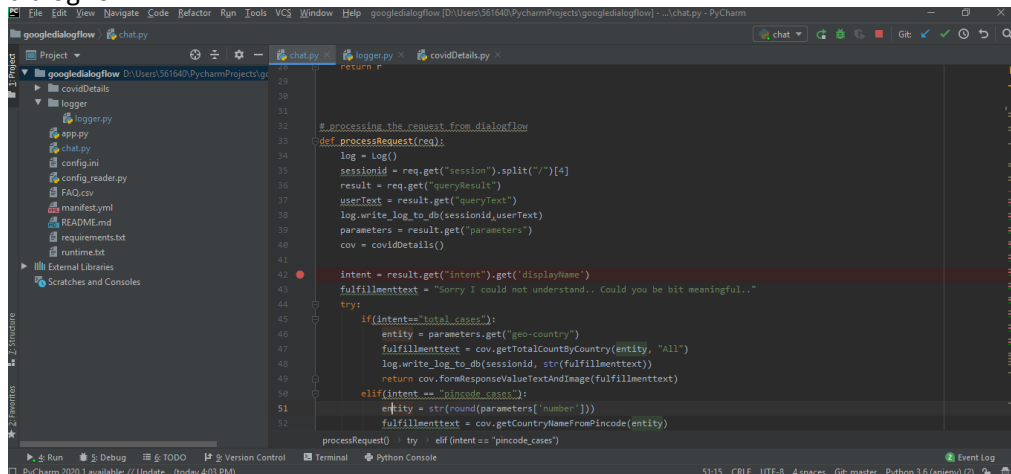


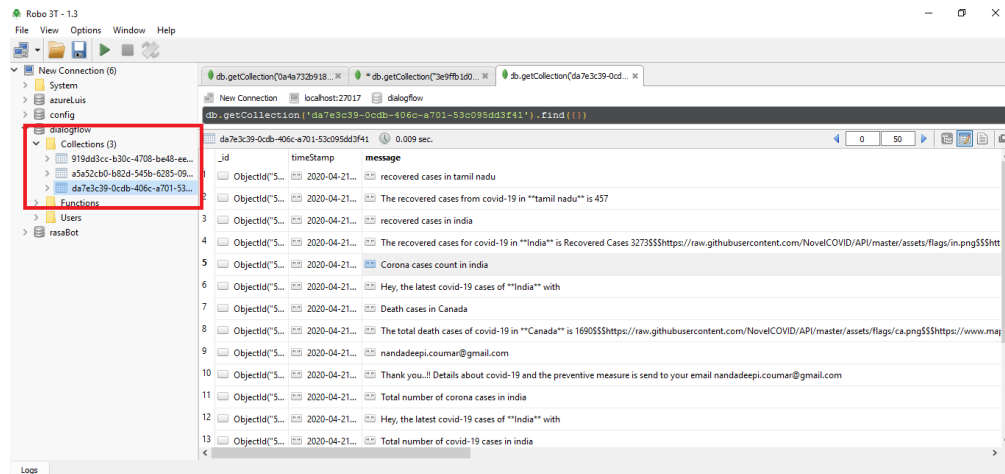


- For the each and every intent created, map the respective responses in cases needed.
- Enable the fulfilment section for the respective intents needed.
- Under the webhook section configure the ngrok https rest api that has to be invoked and send the necessary details.

6.3 Python Project Setup:

- Create a project “covid_dialogFlow”, using pycharm.
- Create app.py where the Flask based REST api’s will be created.
- From the POST request get the appropriate intent and the entity.
- Create the RULE based engine based on the intent and pass the required parameters to the CUSTOM end point.
- Once the response is obtained, form the fulfilment response and send it back to the dialogflow.

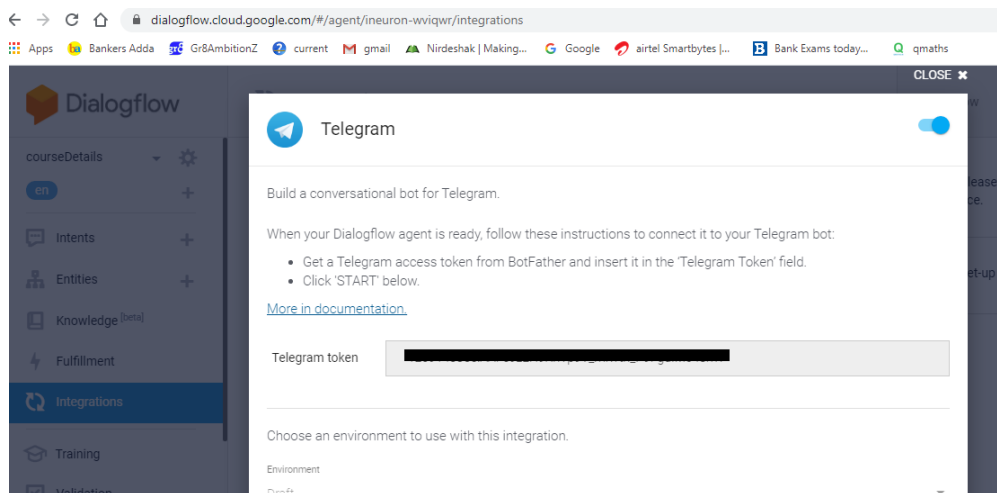




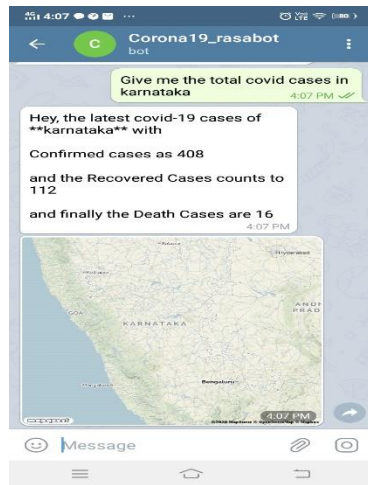
6.4 Telegram Integration:

In order to set up the Telegram integration for your agent, you'll need the following:

- Login to Telegram and go to <https://telegram.me/botfather>
- Click the **Start** button in the web interface or type /start
- Click on or type /newbot and enter a name
- Enter a username for the bot, ending in "bot" (e.g. garthweatherbot)
- Copy the generated access token
- In Dialogflow, go to **Integrations** in the left hand menu
- Click on the **Telegram** title
- Paste the **Access Token** into the related field

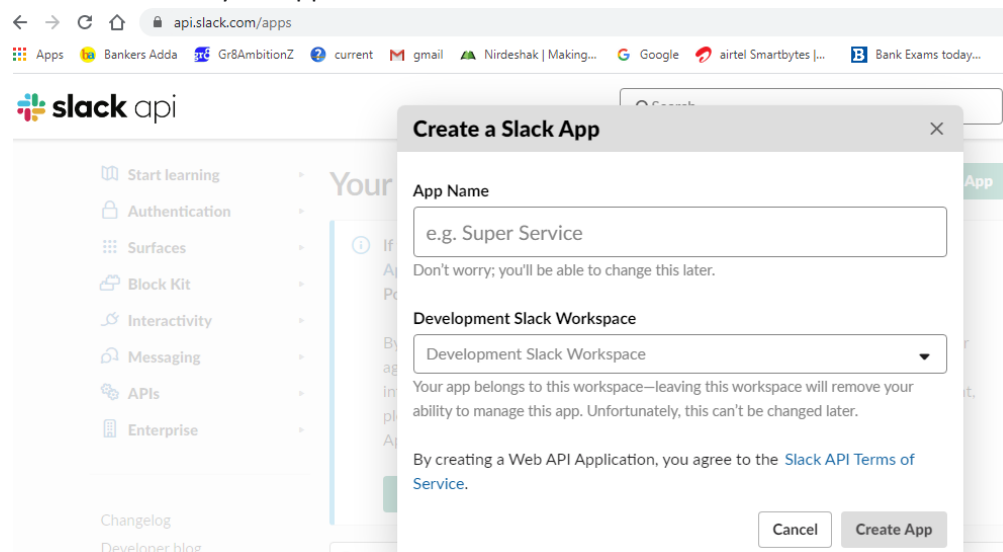


- Click the **Start** button and Now the bot integrated with Telegram.



6.5 Slack Integration:

- Navigate to the [Slack Developer Console](https://api.slack.com/apps)
- Enter a name for your app



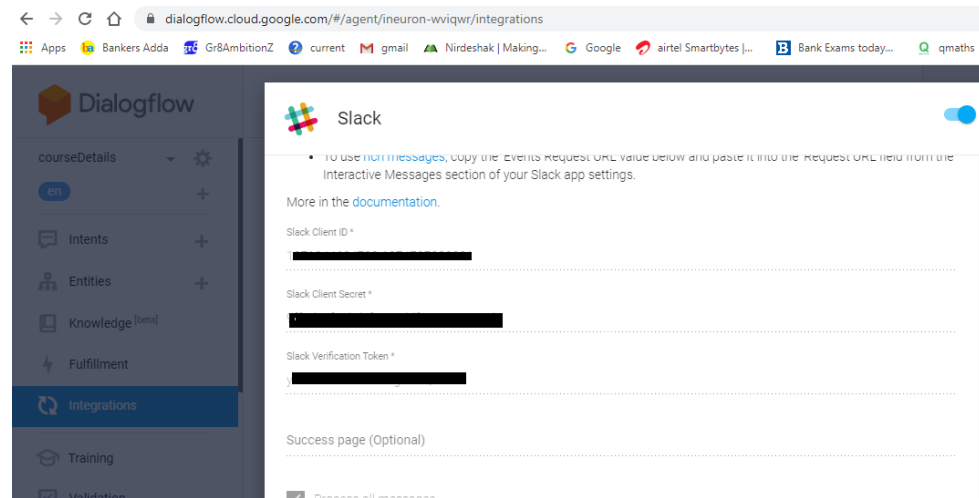
- Choose a Team you would like the app associated with
- Click the **Create App** button

6.5.1 Enabling Integration in Dialogflow

- In the Slack Developer Console, click on **Basic Information** in the left hand menu and scroll down to the **App Credentials** section. Make note of the **Client ID**, **Client Secret**, and **Verification Token**.
- 6.5.2 Link Slack to Dialogflow
- In Dialogflow, go to **Integrations** in the left hand menu
- Click on the **Slack** tile
- Enter the related values into the following fields:
 - **Client ID**
 - **Client Secret**

Verification Token

- Click **Start**



6.5.3 Add Event Request URL

- Copy the **Event Request URL** from Dialogflow's Slack setting page
- Go to the Slack Developer Console and click on **Event Subscriptions** in the left menu
- Click on the switch to enable Events, and paste the Event Subscription URL in to **Request URL** field

6.5.4 Enable Event Subscriptions

- Click on **Event Subscriptions** in the left menu
- Click the switch to enable the option
- Under the **Subscribe to Bot Events**, click on the **Add Bot User Event** button and add the below events

Event Name	Event Description
message.im	A message was posted in a direct message channel
message.groups	A message was posted to a private channel
message.channels	A message was posted to a channel
im_created	A DM was created

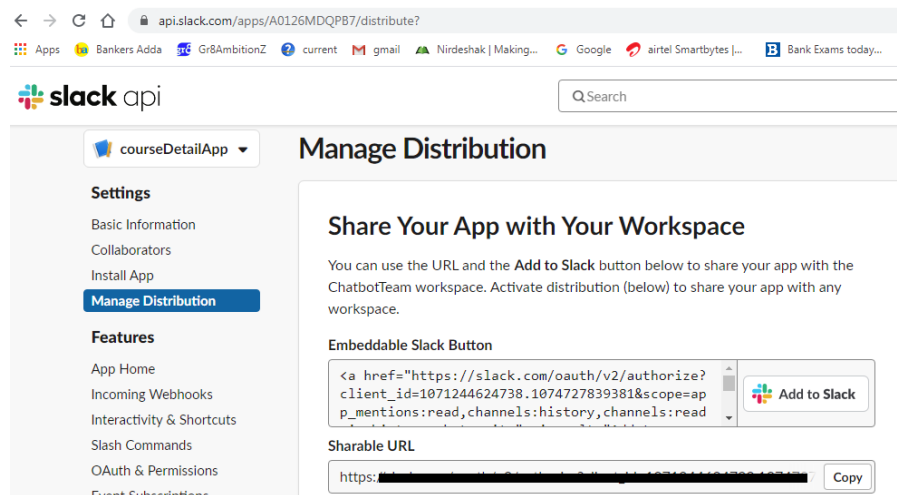
- Once you've added the events, click the **Save Changes** button at the bottom of the screen

6.5.5 Add OAuth URL

- Copy the **OAuth URL** from Dialogflow's Slack setting page
- Go to the Slack Developer Console and click on **OAuth & Permissions** in the left hand menu
- Click on the **Add a new Redirect URL** button and paste the OAuth URL
- Click **Add**

6.5.6 Add Your Slack Bot to a Team

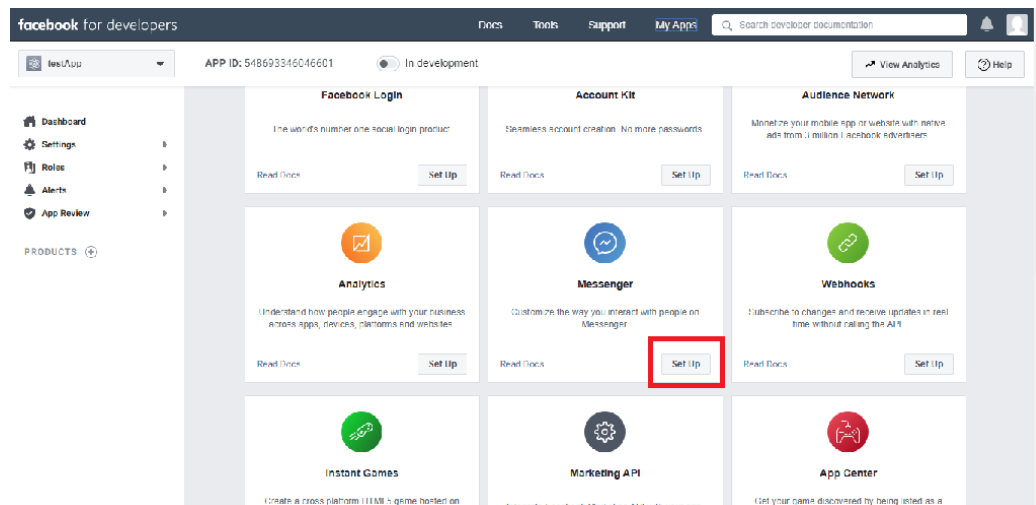
- In the Slack Developer Console, click on **Manage Distribution** in the left menu
- Under the **Share Your App with Your Team** section, click the **Add to Slack** button



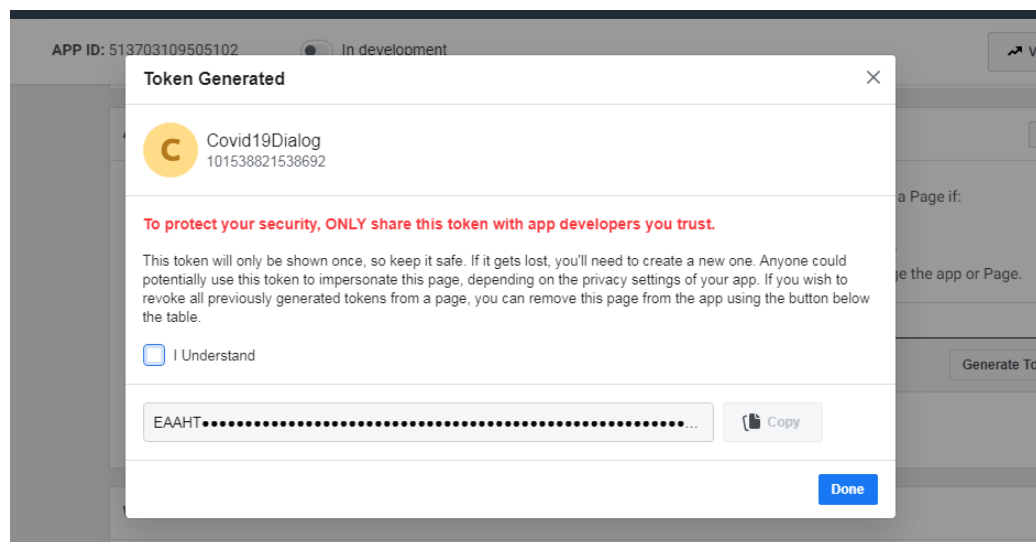
- On the next page, click the **Authorize** button to add the agent
- Now the bot integrated with slack channel.

6.6 Facebook Integration:

- Login to the FB application and create a page.
- Login to the FB for developers, and under MyApps Click "Create APP"
- After Creating App, select Messenger -> Setup.

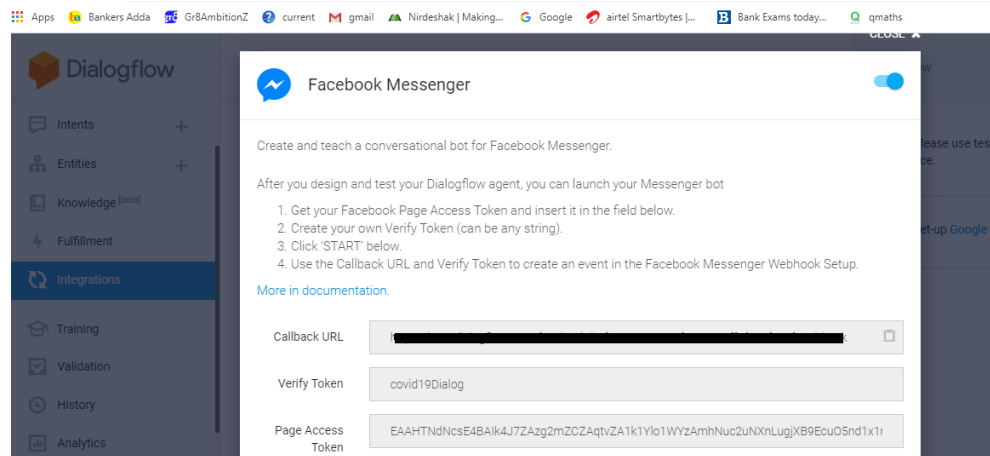


- Under Access Tokens, Add you app and generate the token.



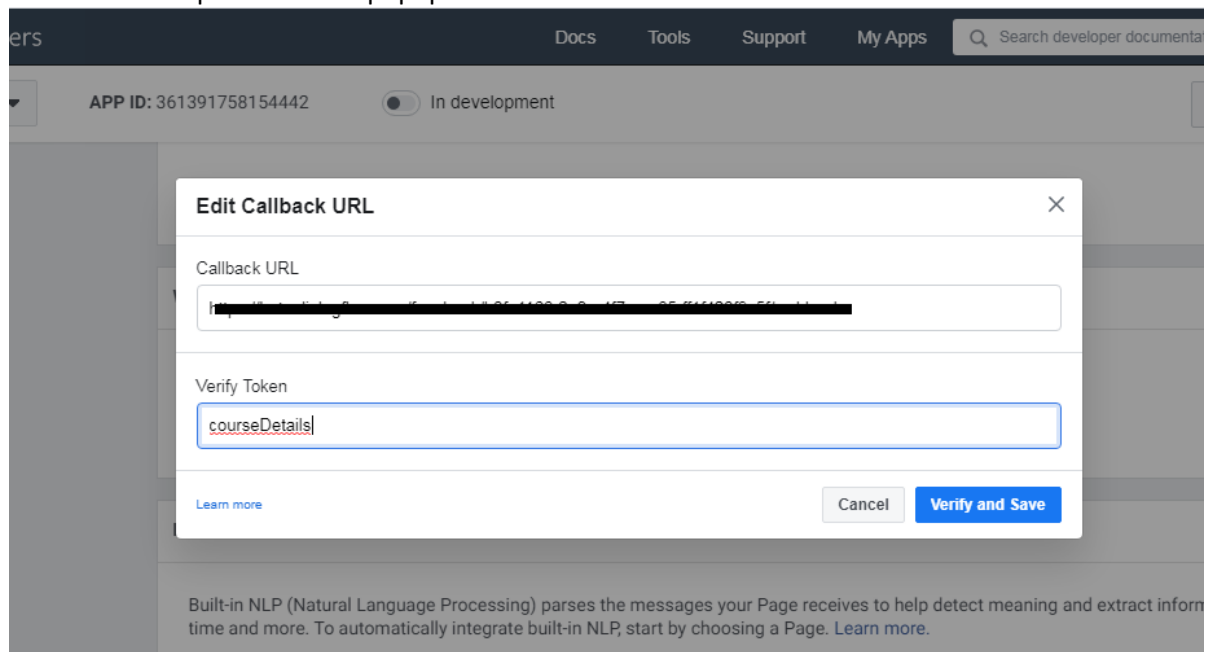
6.6.1 Enable Facebook Messenger Integration in Dialogflow

- To publish Dialogflow agent on our Facebook app, we need to enable Facebook Messenger integration under Integration tab of Dialogflow.
- Clicking on Toggle of Facebook Messenger will open up a popup. In this popup we need to provide the Verify Token which can be any random string. We also need to provide the Page Access Token of Facebook App which we generated in earlier steps.
- After filling up Verify Token and Page Access Token click on “START” button which will activate the Dialogflow webhook for our agent.



- We are done with configuration of Dialogflow agent. Now, we need to setup webhook in Facebook App which will communicate with Dialogflow Agent.

- To setup webhook, open Settings page of Facebook App and click on “Setup Webhooks” button.
- Clicking on Setup Webhooks will open up a popup window where we need to input Callback URL, Verify Token and select Subscriptions which we will require for our bot.
- To copy Callback URL and Verify Token from the Dialogflow agent, open Integrations page and click on Facebook Messenger. That will provide us with the required values. Copy from there and paste it in the popup.



- For Subscription Fields, “messages” and “messaging_postbacks” are sufficient to get started with basic bot. So, we select only those 2 for now.

- Next step in Webhook Setup is selecting the Facebook Page to listen to the events. Select the page from Dropdown and click Subscribe.
- Open your Facebook Page and click on Add A Button.
- It will open a popup. In first step, select “Send Message” and click Next
- In second step, select “Messenger” and click Finish.
- You will see “Send Message” button on the page now. Your visitors can click it and start chatting with your page. As you are admin, you will need to mouse over the “Send Message” button and click on “Test Button” option.

